NSF AWARD #1835822

An Extensible Geospatial Data Framework (GeoEDF) for FAIR Science

Carol Song
Rajesh Kalyanam

Purdue GIS Day
NOVEMBER 07, 2019

The GeoEDF Project

An Extensible Geospatial Data Framework Towards FAIR Science

To help data-driven sciences to be more
Findable, Accessible, Interoperable, Reusable

funded by NSF CSSI program (Cyberinfrastructure for Sustained Scientific Innovation), Data Framework track, $4.5M

October 2018 - September 2023

Project Leadership

Jian Jin

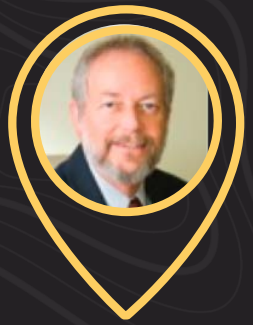
Plant phenotyping & sensors

Venkatesh Merwade

Flood modeling & visualization

Carol Song

PI
Cyberinfrastructure



Jack Smith

Water Quality & resource management

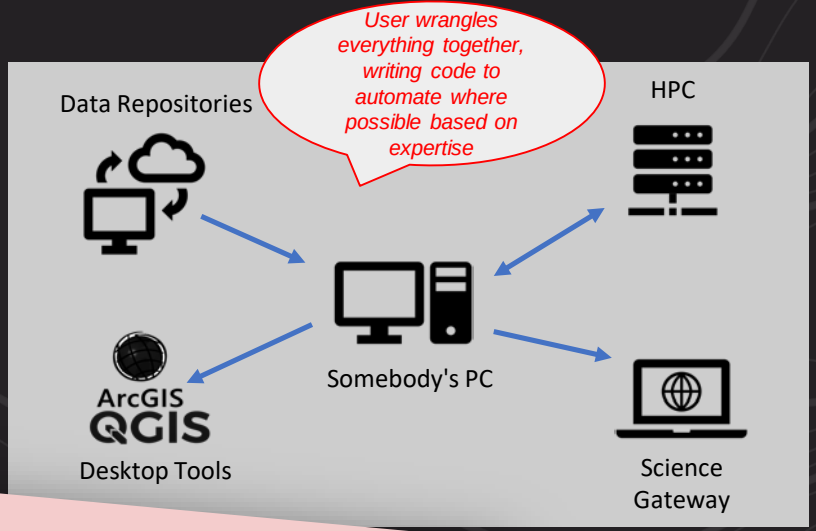
Uris Baldos

Sustainable development

Wrangling of data, computation, software, …

OUR DATA WORKFLOW - Ver. ~~1~~ ~~2~~ 3

1. Make sure date is just after 1st or 15th!
2. Go to: ~~usgs.gov~~ prd-tnm.s3.amazonaws.com/index.html?prefix=StagedProducts
3. Browse: Hydrography…NHDPlusHR…Beta…GDB.
4. Download NHDPLUS_H_01##_HU4_GDB.zip where ## is **02 to 14**.
5. Unzip it - WARNING: Have enough space!!!
6. Run our tool. WARNING: Takes a **looooong** time!!! DO NOT TURN OFF PC!!!
7. Upload output files to cluster - Note: wait until all successful!
8. Kick off our standard jobs.
9. ~~Occasionally check 'em~~. Wait for email(s)???
10. Download new images.
11. Ask ~~Fred~~ to upload to website.
12. Tell everybody there's new stuff. Mary?



Process your files before running our tool

- get the latest code from ~~Jaewoo~~ Nicole for filtering the input data

Note: that's Windows code – use the lab desktop

- you need to get the maps from the group folder /depot/lyllegroup/project1/maps/exp1/ …. for aggregation

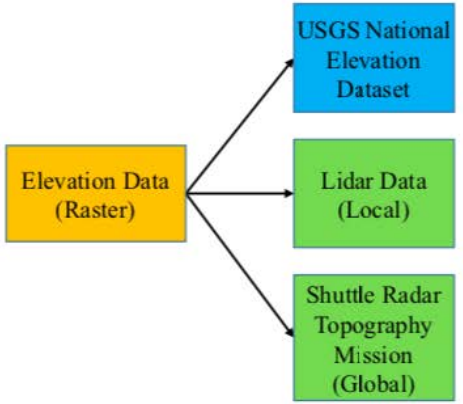
Capturing complex data pipelines (example)

Potential pitfalls with other elevation data sources

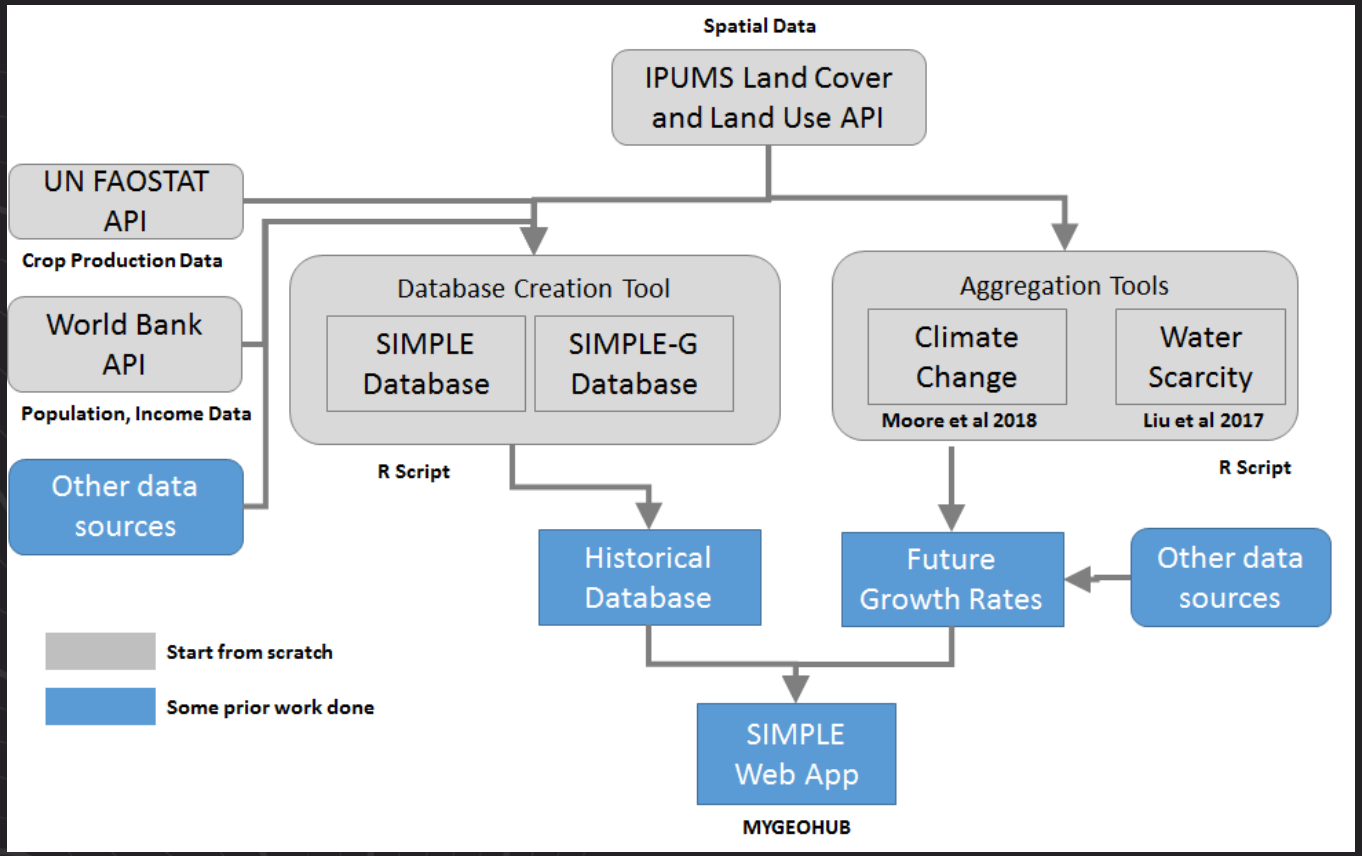
- 1/9th arc-second or LiDAR or 1-m DEM not available nationally
- Pre-processed Lidar available in some states (IN, OH, MN, NC)
- Create Lidar acquisition tool where available or not?
- Conversion of Lidar point cloud to bare-earth DEM is an issue
- SRTM → available globally at different resolutions
- Probably need to re-project user shapefile to USGS coordinates first

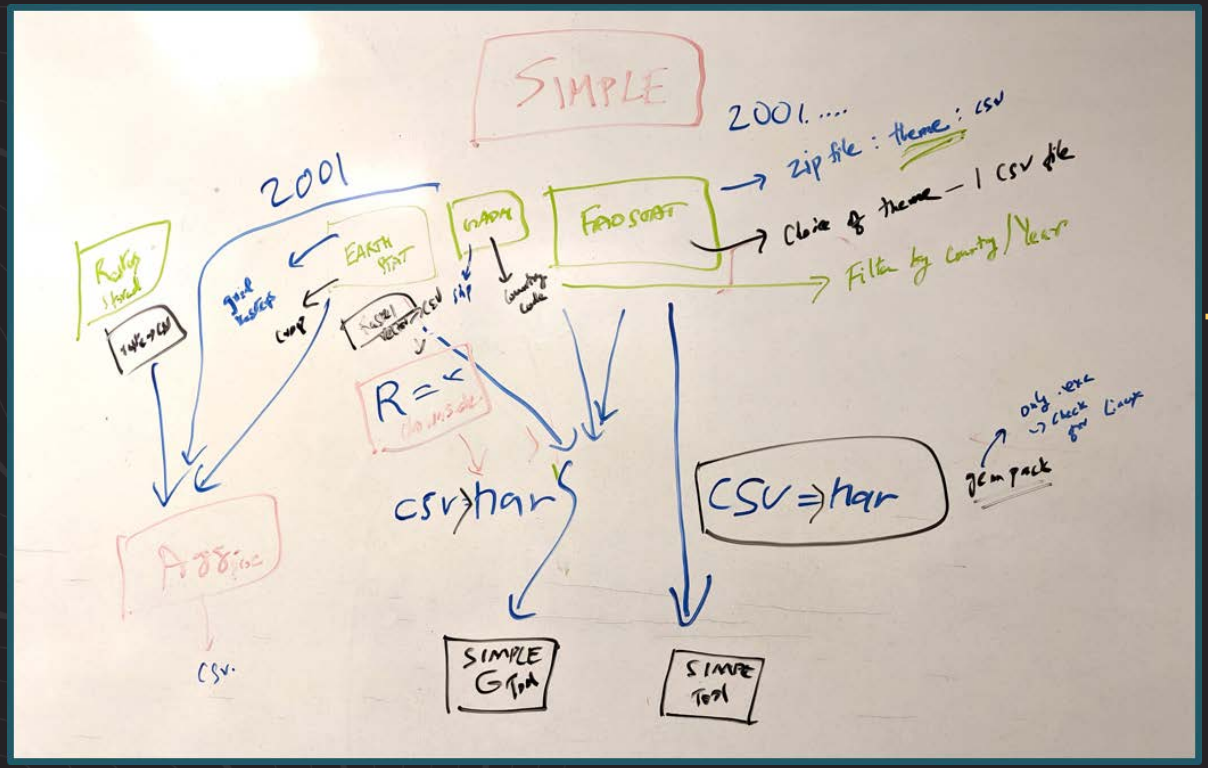
1-m DEM availability

Lidar Point cloud availability



Example Workflow





There are a lot more details

GABBs 2.0: GeoEDF -- Vision

Create an extensible geospatial data framework that will address the challenges by providing seamless connections among platforms, data and tools, hence making valuable, large scientific and social datasets usable directly in scientific models and tools.

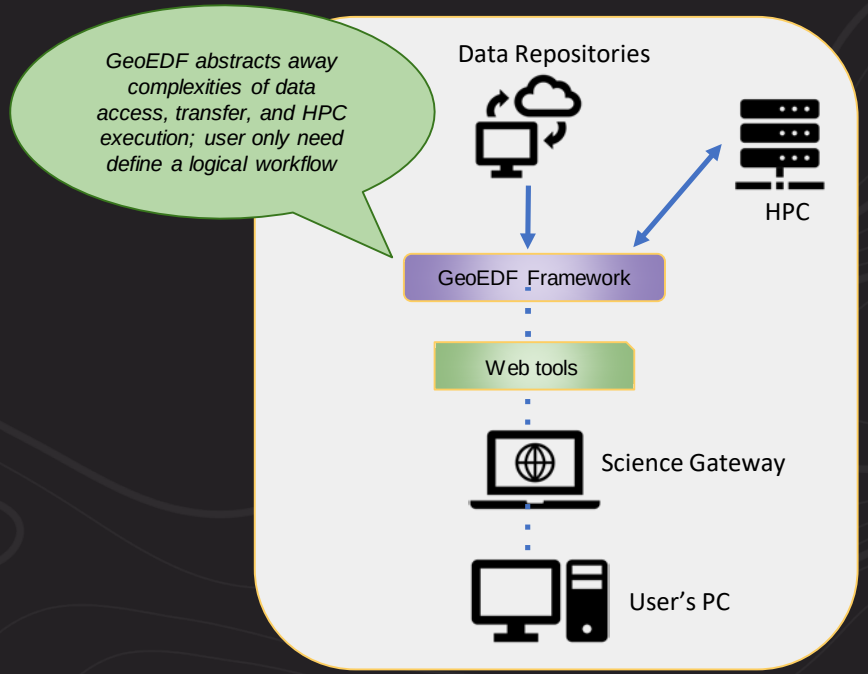
The ultimate goal is to put easy-to-use tools and platforms into the hands of researchers and students to conduct scientific investigations following FAIR science principles.

FAIR = Findable, Accessible, Interoperable, Reusable

Vision: After GeoEDF

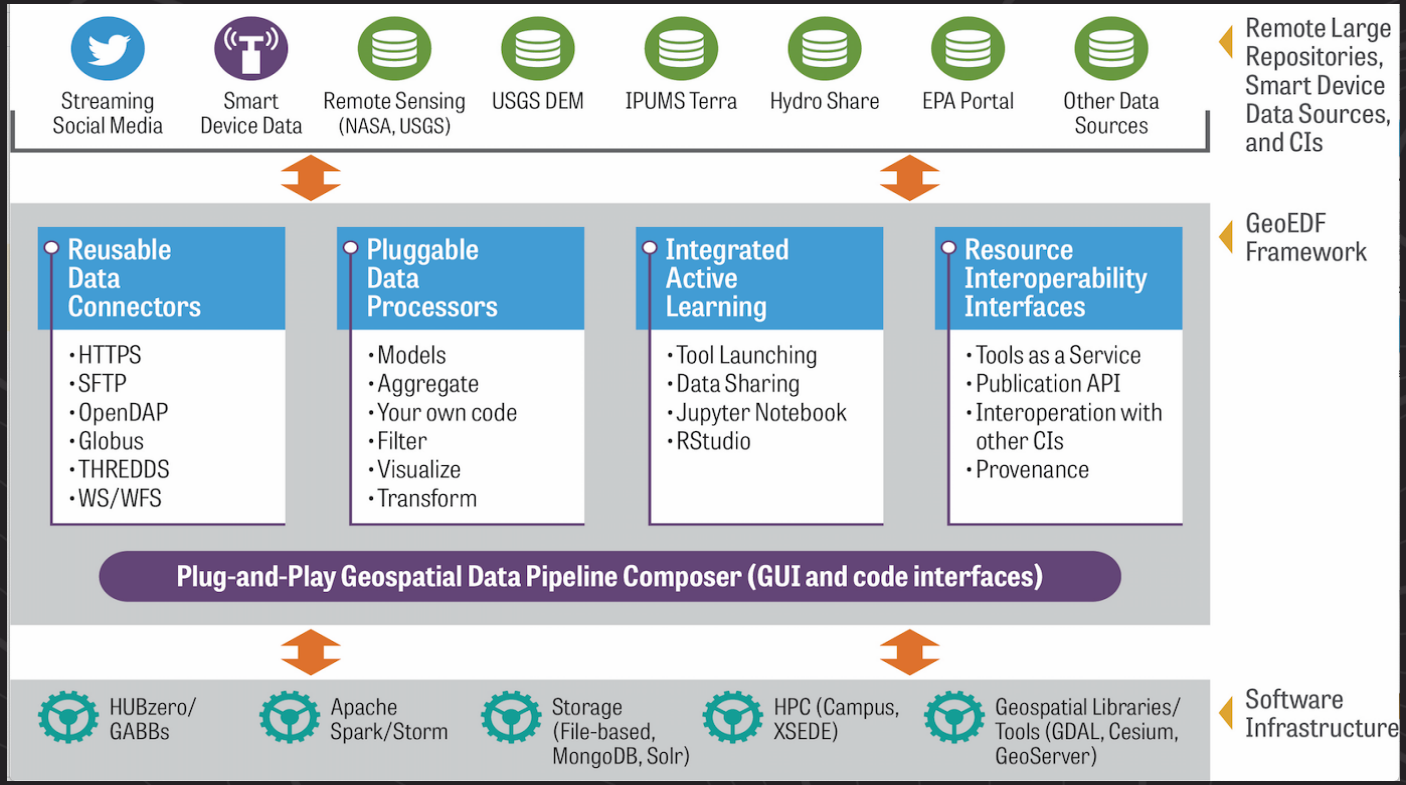
OUR DATA WORKFLOW - Final

1. *Go to the science gateway*
2. *Define "my_workflow.yml" (or use tool GUI if needed)*
3. *Ask GeoEDF to execute!*
4. *Data and workflow automatically published to science gateway*



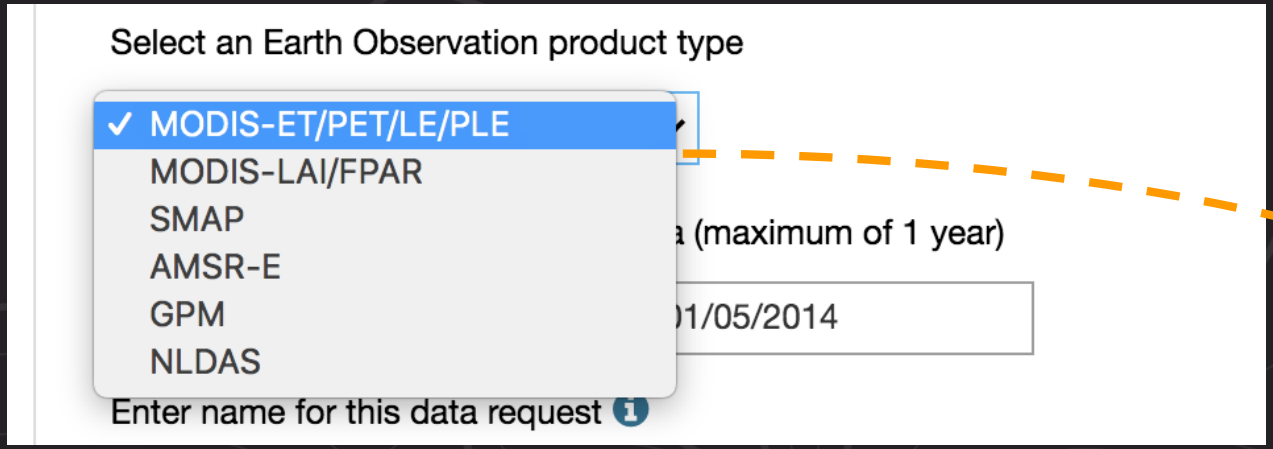
- Automated, secure, logged process running on dedicated infrastructure - You can log off!
- Leverage building blocks from existing workflows
- Data transfer and HPC execution abstracted away
- Automatic provenance capture and data annotation for future discoverability, reproducibility

GeoEDF High-Level View



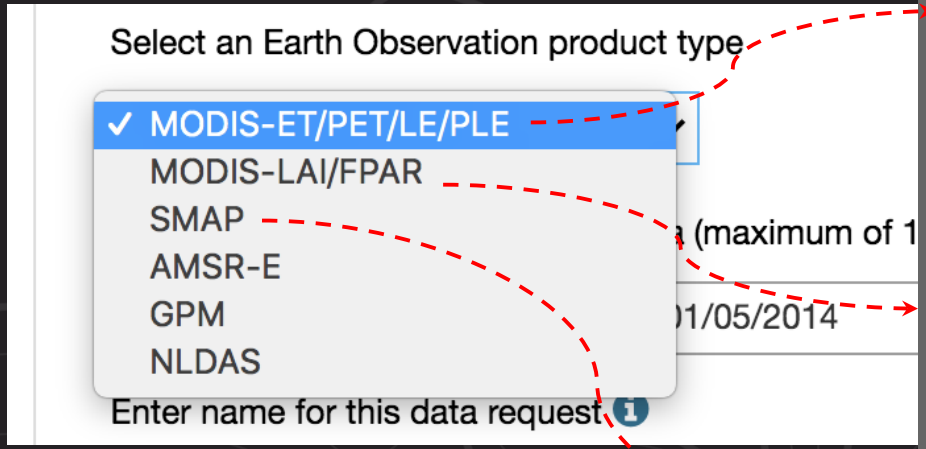
Design and Cyberinfrastructure

Workflow Example I



Mask with shapefile, compute weighted aggregate for each polygon

Workflow Example I - Opportunity

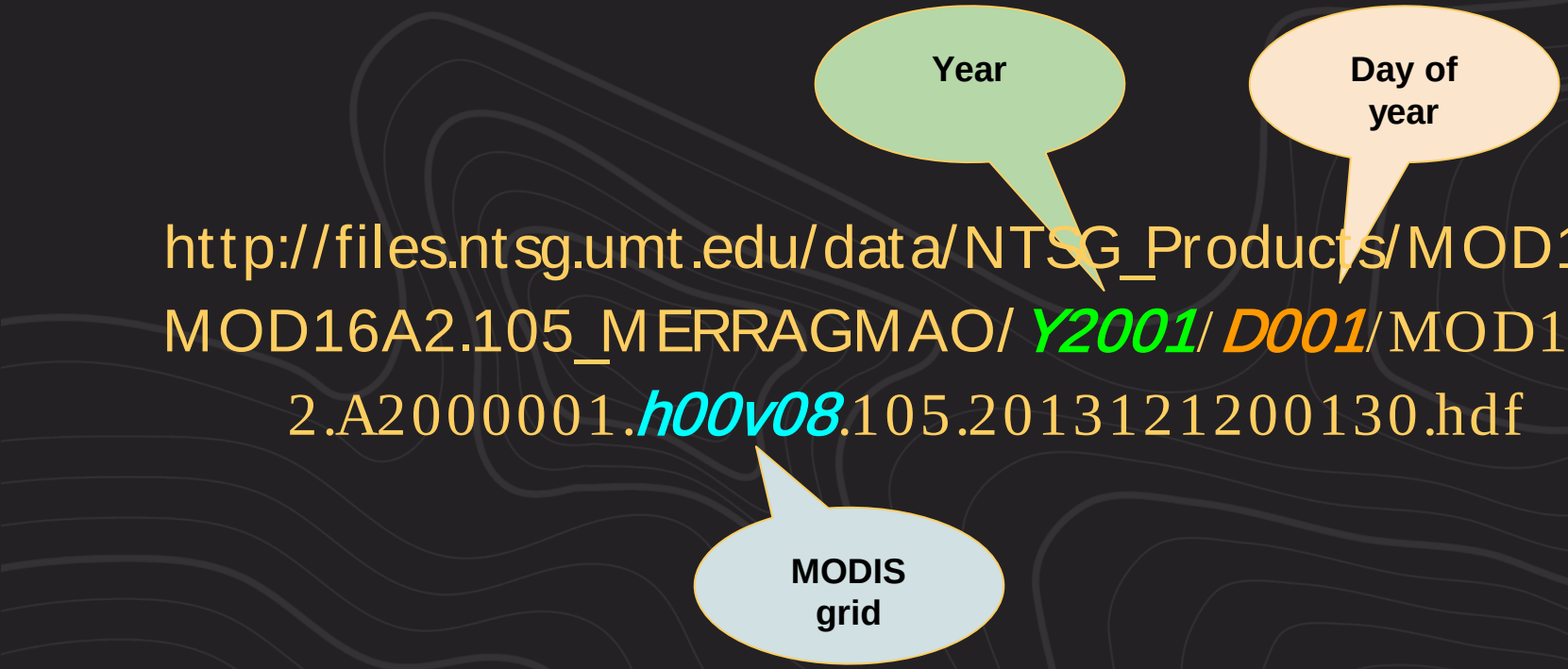


http://files.ntsg.umt.edu/data/NTSG_Products/MOD16/MOD16A2.105_MERRAGMAO/Y2001/D001/MOD16A2.A2000001.h00v08.105.2013121200130.hdf

https://e4ftl01.cr.usgs.gov/MOTA/MCD15A3H.006/2002.03.19/MCD15A3H.A2002193.h07v07.006.2015149100709.hdf

https://n5eil01u.ecs.nsidc.org/SMAP/SPL4SMGP.003/2015.03.31/SMAP_L4_SM_gph_20150331T013000_Vv4030_001.h5

Workflow Example I – Opportunity (continued)



Workflow Example II

```
def GetNED(NL, WL):
    name1 = "n"+NL+"w"+WL
    address = "ftp://rockyftp.cr.usgs.gov/vdelivery/Datasets/Staged/Elevation/1/ArcGrid/USGS_NED_1_"
    url_final = address + name1 + "_ArcGrid.zip"
    print(url_final)
work_folder_name = os.path.join(input_folder_name, "WorkFolder")
if os.path.exists(work_folder_name) == False:
    os.mkdir(work_folder_name)
boundary_path = os.path.join(input_folder_name, boundary_file)
input_crs = QgsVectorLayer(boundary_path, '', 'ogr' ).crs().authid()
#processing.run('qgis:reprojectlayer',{'INPUT': full_input_path, 'TARGET_CRS':'EPSG:10267
processing.run('native:reprojectlayer',{'INPUT': boundary_path, 'TARGET_CRS':'EPSG:4326',

input_list = [os.path.join(work_folder_name, cur_raster) for cur_raster in raster_names]
print("Merging Raster...")
processing.run("gdal:merge", {'INPUT':input_list, 'OUTPUT':work_folder_name + "/merged_rast.tif"})
print("Projecting Raster...")
processing.run('gdal:warpreproject', {'INPUT': work_folder_name + "/merged_rast.tif", 'TARGET_CRS':
print("Clipping Raster...")
processing.run('gdal:cliprasterbymasklayer',{'INPUT': work_folder_name + "/proj_rast.tif", 'MASK':
print("DEM prepared successfully!!!")
```

Workflow Example II - Opportunity

Get NED from USGS

```
def GetNED(NL, WL):
    name1 = "n"+NL+"w"+WL
    address = "ftp://rockyftp.cr.usgs.gov/vdelivery/Datasets/Staged/Elevation/1/ArcGrid/USGS_NED_1_"
    url_final = address + name1 + "_ArcGrid.zip"
    print(url_final)
```

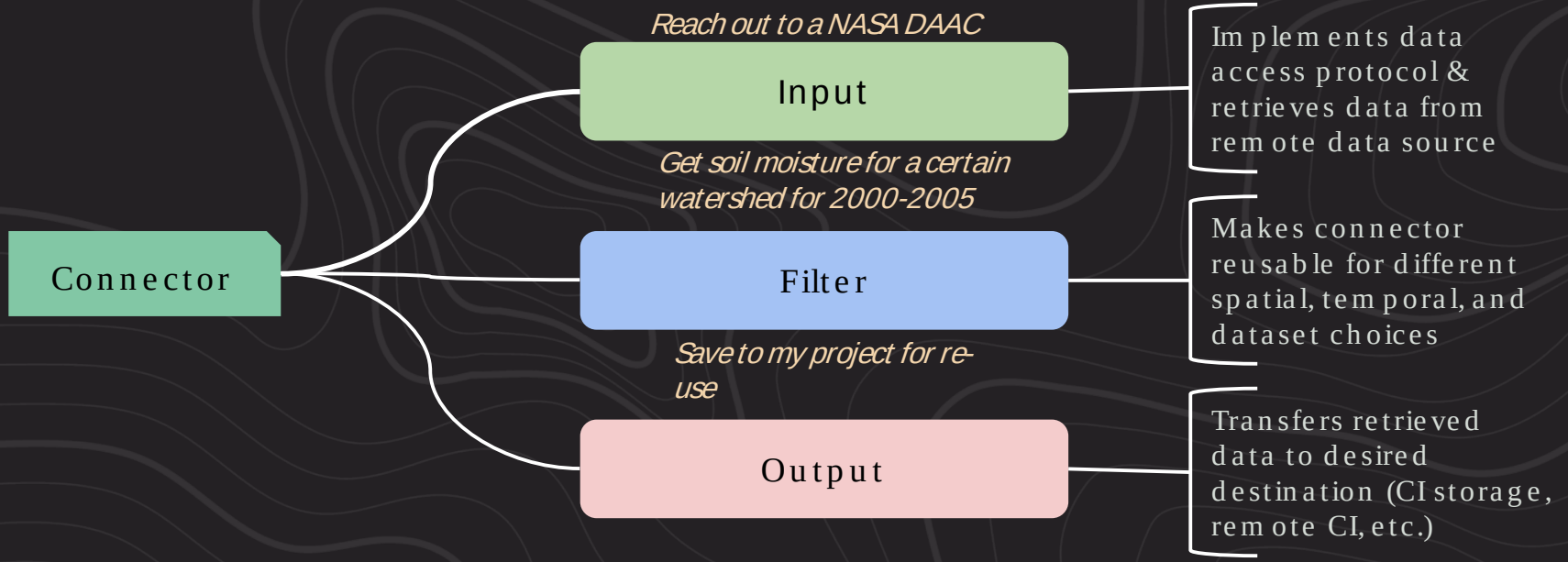
Reproject watershed shapefile

```
work_folder_name = os.path.join(input_folder_name, "WorkFolder")
if os.path.exists(work_folder_name) == False:
    os.mkdir(work_folder_name)
boundary_path = os.path.join(input_folder_name, boundary_file)
input_crs = QgsVectorLayer(boundary_path, '', 'ogr' ).crs().authid()
#processing.run('qgis:reprojectlayer',{'INPUT': full_input_path, 'TARGET_CRS':'EPSG:10267
processing.run('native:reprojectlayer',{'INPUT': boundary_path, 'TARGET_CRS':'EPSG:4326',
```

Mosaic -> reproject -> clip raster(s)

```
input_list = [os.path.join(work_folder_name, cur_raster) for cur_raster in raster_
print("Merging Raster...")
processing.run("gdal:merge", {'INPUT':input_list, 'OUTPUT':work_folder_name + "/merged_rast.tif"})
print("Projecting Raster...")
processing.run('gdal:warpreproject', {'INPUT': work_folder_name + "/merged_rast.tif", 'TARGET_CRS':
print("Clipping Raster...")
processing.run('gdal:cliprasterbymasklayer',{'INPUT': work_folder_name + "/proj_rast.tif", 'MASK':
print("DEM prepared successfully!!!")
```

Data Connectors - Design



Data Connectors - Example Definition

```
Input:
    NASAInput:
            url:

http://files.ntsg.umt.edu/data/NTSG_Products/MOD16/MOD16A2.105_MERRAGMAO/%{file}
            user: rkalyana
            password:
Filter:
    file:
        PathFilter:
            pattern: 'Y%{year}/D001/*.h00v08*.hdf'
    year:
        DateTimeFilter:
            pattern: '%Y'
            start: 01/01/2000
            end: 12/31/2005
            period: 1Y
```

Python Input class

Variable to be bound by filter

Filter returns string bindings for variable

Filter params can also have variables

Data Processors - Example Definition

Python processor class implementing masking operation

```
HDFShapefileECOSMask:
        hdffile: /data/workflow263/mod16Y2001D1T1200.h00v08.hdf
        shapefile: /home/rkalyana/subs1.shp
```

Processor specific params; validated during instantiation

GeoEDF Workflow - Example Definition

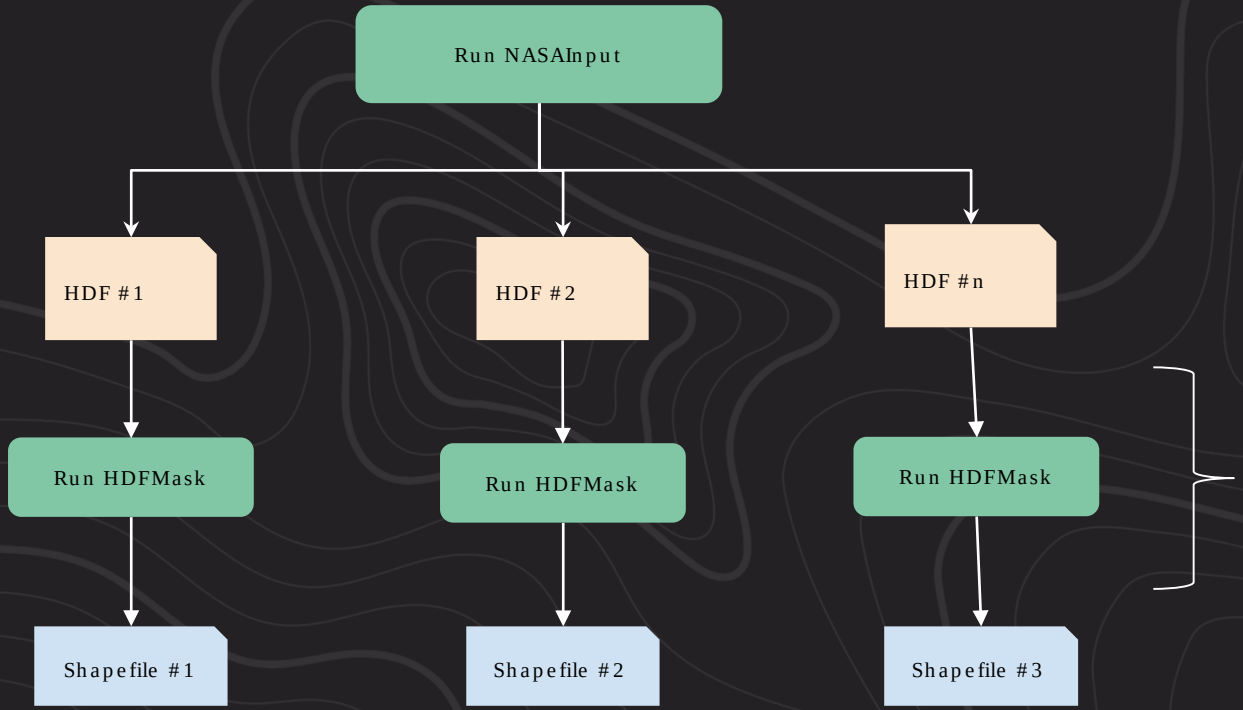
```
$1:
  Input:
    NASAInput:
      url:
        http://files.ntsg.umt.edu/data/NTSG_Products/MOD16/MOD16A2.105_MERRAGMAO/%{file}
      user: rkalyana
      password:
  Filter:
    file:
      PathFilter:
        pattern: 'Y%{year}/D001/*.h00v08*.hdf'
    year:
      DateTimeFilter:
        pattern: '%Y'
        start: 01/01/2000
        end: 12/31/2005
        period: 1Y
$2:
  HDFShapefileEOSMask:
    hdffile: $1
    shapefile: /home/rkalyana/subs1.shp
```

Workflow stage

Reference output of prior stage

- ★ NASAInput can be used to access any Earthdata-associated repository
- ★ HDFShapefileEOSMask can be applied to any HDF4 or HDF5 file

Actual Scientific Workflow



Parallel compute jobs generated dynamically based on number of data files at each level

Putting It All Together

Workflow Definitions

Users pick and choose different connector & processor classes to define a workflow

(as YAML file/via GUI/through API)

Workflow Execution

Workflow engine transforms declarative specification into concrete Pegasus scientific workflow and executes on heterogeneous compute

GeoEDF Building Blocks

Users contribute various connector (Input, Filter, Output), and processor classes

FAIR Science

❖ How do we do FAIR?

- ➢ Data publications can be searched using their content metadata, accessed via APIs & used in workflows
- ➢ Automatically track metadata, provenance in workflows
- ➢ Launch tools, workflows seamlessly from a remote CI (with remote data inputs)

Questions